UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/714,804	11/16/2000	Richard Shann	S1022/8572	3979

7590 09/24/2007

James H Morris
Wolf Greenfield & Sacks PC
600 Atlantic Avenue
Boston, MA 02210

EXAMINER	
ROMANO, JOHN J	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
09/24/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mN

Office Action Summary	Application No.	Applicant(s)	
	09/714,804	SHANN ET AL.	
	Examiner	Art Unit	
	John J. Romano	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 June 2007 + 6/04/2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-8 and 12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-8 and 12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's amendment and response received June 04th, 2007 and supplemental response received June 18th, 2007 responding to the December 12th, 2006, Office action provided in the rejections of claims 1-8. The June 04th, 2007 response amended claims 1, 4-8 and added new claims 9-12. Subsequent to the interview of record, which took place on June 12th, 2007, and mailed June 21st, 2007, the Applicant submitted the amendment and response received June 18th, 2007 supplementing the previous response. In the supplemental response, claims 1, 2, 5-8 and 12 are amended, claims 9-11 are cancelled. Accordingly, claims 1-8 and 12 are pending in the application and which have been fully considered by the examiner.

In regard to Applicant's arguments pertaining to the June 04th, 2007 response, it is noted that said response was previous to the interview, which took place on June 12th. However, since Applicant did not submit arguments with the supplemental response rather only remarks noting the supplemental amendments, the examiner addresses the arguments of the June 04th, 2007 response to the extent that they appear consistent with the supplemental claim language. However, it is noted that the arguments of the June 04th, 2007 response are in regard to the now cancelled claim language of the corresponding dated claim limitations.

As noted in the Examiner's Interview Summary, mailed June 21st, 2007, Applicant's representatives discussed the invention and differences of the invention over the prior art of record and in regard to the claims as submitted in the June 04th,

2007 claim listings. In particular, Applicant's representative pointed out the ability to constrain the machine language rather than the assembly language, thereby allowing the machine language to conform to the instruction set (Applicant's Drawings, Figure 2, 30) constraints provided in the description file (24). The examiner recommended amending the claims in the direction of the flexibility or adaptability to more precisely convey the differences over the prior art. Subsequently, Applicant's representative submitted the supplemental claim amendments dated June 18th, 2007, and addressed herein-below.

The rejection of the claims over prior art in the previous Office action is maintained in light of additional new grounds of rejection as necessitated by amendment and **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Prior Art's Arguments – Rejections

2. Applicant's arguments filed June 04th and June 18th, 2007, in particular on pages 7-10 of the June 04th, 2007 response, have been fully considered but they are not persuasive. For example,

(A) In regard to the supplemental amended claim limitations, the Examiner notes that Applicant did not point out support for "*automatically track changes in an instruction set*" or "*derived and stored as constraints due to the instruction set*". The examiner looked to the specification and found the following passage (See originally filed specification, page 9, lines 7-12):

> "The source code is also input to a fetch unit 23 whose function is to address a descriptor file 24 with information derived from the source code currently being translated to provide output information 25 **representative of constraints due to the instruction set architecture**." (emphasis added).

Accordingly, the examiner interprets tracking changes in an instruction set as the instruction set output information (25) being **representative of constraints due to the instruction set architecture** provided by the Instruction Set Description Language (**Hadjiyiannis**), or respectively provided by the Description File 24, (Applicant), wherein both provide instruction set information to generate changing (retarget-able) assembly code. Herein, Applicant's flexibility is achieved by tracking changes to an instruction set 30 by providing the descriptor file 24 as input (Figure 2) during generation of executable code 50 (page 9, lines 31-35). It appears that tracking the changes is actually

incorporating the changes of the instruction set (e.g., different instruction set, format, encoding function, etc.), thereby representing the constraints due to the instruction set. Similarly, **Hadjiyiannis** flexibility (retarget-ability) is achieved by varying the machine description including an instruction set specification. It is also noted that this is consistent with Applicant's provided support for the claim amendments in the June 4th, 2007, response (See page 6), wherein Applicant cites page 10, lines 18-30 for support to the claim amendments of the respective response.

(B) Applicant's remaining arguments have been considered but are moot in view of the new ground(s) of rejection necessitated by amendment.

Claim Rejections

Claims 1-8 and 12, are pending claims, stand rejected in light of the additional clarifications provided and/or addressed at item 2 above, Prior Art's Arguments – Rejections.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims **1-8** and **12** rejected under 35 U.S.C. 103(a) as being unpatentable over Hadjiyiannis et al, "ISDL: An Instruction Set Description Language for Retargetabillity", (hereinafter **Hadjiyiannis**) and further in view of Vos, GB 2,127,188A (art of record and hereinafter **Vos**) and further in view of A.N.Edmonds "Microcoding and bit-slice techniques" (art of record & hereinafter **Edmonds**).

In regard to claim **1**, **Hadjiyiannis** discloses:

- "An assembler for a target microprocessor to automatically track changes in an instruction set of the target microprocessor, the assembler comprising..." (E.g., see Fig. 1 & Section I, Paragraph C, lines 1-9), wherein the ISDL is varied according to the target processor, the varied/changed ISDL defines the changed instruction set, thereby automatically incorporating/tracking the changes in an instruction set of the target. It is noted that tracking the changes is interpreted as being **representative of constraints due to the instruction set architecture** provided by the Instruction Set Description Language (**Hadjiyiannis**), or respectively provided by the Description File 24, (Applicant), wherein both provide instruction set information to generate changing (retarget-able) assembly code and as further addressed in section (A) above.
- "*...a descriptor file containing information descriptive of the instruction set of said target microprocessor...*" (E.g., see Fig. 1 & Section I, Paragraph C, lines 1-9), wherein the Architecture Synthesis System

comprises the machine description including an ISDL (Instruction Set Description Language) specification and some architectural information.

- "...*wherein information about ...at least one bit field and a number of bits available for the at least one bit field for at least one instruction in the instruction set is derived and stored as constraints due to the instruction set*..." (E.g., see Fig. 1 & Section III,), an ISDL description comprises an instruction word format, instruction set, constraints and optional architectural details. See Section III, (A), where the instruction word format section defines the hardware instruction word by providing the subfield and bit-width of the instruction word. Also, see Section "IV. An ISDL Example", wherein the section format defines the number of bits (width) of the bit-field for one instruction in the instruction set of the "DSP56000 processor to illustrate the structure of ISDL, and how it is used".
- "...*a translation device of the assembler for translating assembly language instructions into machine language as an output; a fetching device of the assembler for deriving information from the assembly language instructions and acquiring data from said descriptor file...*" (E.g., see Fig. 1 & Section III, Paragraph 1, lines 2-11), wherein the Architecture Synthesis System transmits the ISDL description to the compiler. The compiler outputs machine specific assembly code,

which is translated to machine language via the automatic assembler generated by the ISDL description. Thus, the translation device comprises the Architecture Synthesis System, the compiler, ISDL description and the ISL. Furthermore, the compiler/assembler fetches the ISDL description from the Architecture Synthesis System.

- "*...a control device arranged to receive said data from said fetching device...*" (E.g., see Fig. 1 & Section III, Paragraph 1, lines 5-7), wherein the compiler is the control device which receives fetches said data and constrains the data to produce code specific to the target processor or instruction set.

But **Hadjiyiannis** does not expressly disclose "*...and said machine language from said translation device, and operable to constrain the machine language to conform to the architecture of said instruction set.*" Instead **Hadjiyiannis** teaches receiving specific details of the instruction set and constraining the assembly language, thereby constraining the machine language as well, to the architecture of said instruction set. Accordingly, it is noted by the examiner, that the compiler performs some of the Applicant's claimed assembler functions. Correspondingly, it is the examiner's position that the compiler is part of the claimed assembler in the sense that the respective mapped functions (i.e., translate, acquiring data from the instruction set) and result (machine language) are equivalent as claimed. Thus, it would have been obvious to one of ordinary skill in the art, to constrain the machine language to conform to the architecture of said instruction set, instead of restraining the assembly language

to conform to the architecture, and thereby restraining the machine language, as the two methods produce the same result.

But **Hadjiyiannis** does not disclose expressly "*...a data transfer device arranged to output the selected data fetched from said descriptor file directly to a linker...*" *However,* ***Vos*** *discloses*:

- "*...a data transfer device of the assembler arranged to output the selected data fetched from said descriptor file directly to a linker...*" (E.g., see Figure 1, block10 + block 12 & page 2, lines 21-34), wherein the linker command file (block 12) is generated for the particular prototype processor and the configuration object file (block 10) includes interrupt vectors and procedures, memory configuration which are interpreted as instruction set information. Furthermore, the configuration file and linker command file are directly input to the linker.

Hadjiyiannis and **Vos** are analogous art because they are both concerned with the same field of endeavor, namely, an architecture that is modifiable by input and adapts source code to such input and correspondingly outputs machine language. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to utilize **Vos'** data capture device in **Hadjiyiannis's** system of claim **1** as an alternate method to implement architectural specifications. The motivation for doing so would have been to have a simpler design for a particular system, where **Vos'** method may be more efficient than **Hadjiyiannis's** for a particular objective.

But **Hadjiyiannis** and **Vos** do not disclose expressly "...*a starting posiiton...*" or "...*using the information, wherein the data is representative of the constraints due to the instruction set and includes selected data comprising at least one encoding function for the at least one instruction due to the changes in the instruction set*..." However, ***Edmonds*** *discloses*:

- "...*a starting position*..." (E.g., see Fig. 1 & "Definition Facilities"), wherein the position and size of a field within the bitmap are disclosed.
- "...*using the information, wherein the data is representative of the constraints due to the instruction set and includes selected data comprising at least one encoding function for the at least one instruction due to the changes in the instruction set*..." (E.g., see Fig. 1 & "Definition Facilities"), wherein Metastep offers different functions providing the opportunity to describe skews and bit-field manipulations in the definitions file and render them transparent to the assembly code. It is noted that the definitions file defines the equivalent of Applicants encoding functions used to produce object code.
- "...*to enable the linker to perform operations on the constrained machine language using the selected data, including operations on external symbols using the at least one encoding function*." (E.g., see Fig. 1 & "Assembler Facilities"), wherein Metastep external hardware configurations forcing the sequencer to the start address or subject to constraints such as having their least-significant four bits made zero.

Hadjiyiannis, Vos and **Edmonds** are analogous art because they are both concerned with the same field of endeavor, namely, an architecture that is modifiable by input and adapts source code to such input and correspondingly outputs machine language. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to utilize **Edmonds'** Meta-assembler in **Hadjiyiannis's** system of claim **1** as an alternate method to implement architectural specifications. The motivation for doing so would have been to have a simpler design for a particular system involving the specification details about the hardware (See **Edmonds**, Figure 1).

In regard to claim **2**, **Hadjiyiannis** discloses:

- "*...wherein the descriptor file comprises syntax information for each instruction...*" (E.g., see Section III, Paragraph 4), wherein the six sections of the ISDL are listed and later further described along with their syntax definitions.
- "*...the control device translated each instruction on the basis of said syntax information.*" (E.g., see Section I, Paragraph C, lines 1-6), wherein the machine description contains the syntax information and the code generator produces code based on that information.

In regard to claim **3**, **Hadjiyiannis** discloses the system of claim **1** discussed above. Furthermore, **Hadjiyiannis** discloses:

- "*A system for assembling a machine language program...*" (E.g., see Fig. 1 & Section III, Paragraph 1, lines 2-11), wherein a binary file

(machine language program), is assembled for the target microprocessor.

But **Hadjiyiannis** does not disclose expressly "*...and further comprising a data capture device having an input for accessing the instruction set of said target microprocessor and having an output, wherein said output comprises said descriptor file.*" *However,* ***Vos*** *discloses*:

- "*...and further comprising a data capture device having an input for accessing the instruction set of said target microprocessor...*" (E.g., see Fig. 1, blocks 2, 4 and 6 & Page 2, lines 8-11), wherein the prompter (data capture device) has an input from the interface requirements, which provides details including the instruction set of a target microprocessor.
- "*...having an output, wherein said output comprises said descriptor file.*" (E.g., see Fig. 1, blocks 2, 4 and 6 & Page 2, lines 11-14), wherein the integration source file is the descriptor file.

Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to utilize **Vos'** data capture device in **Hadjiyiannis's** system of claim **1** as an alternate method to implement architectural specifications. The motivation for doing so would have been to have a simpler design for a particular system, where **Vos'** method may be more efficient than **Hadjiyiannis's** for a particular objective.

In regard to claim **4**, **Hadjiyiannis** discloses the system of claim **1** as described above. But **Hadjiyiannis** dose not disclose expressly "*...a linker wherein the system has a data transfer device outputting selected data fetched from said descriptor file to said linker, whereby said linker uses said output data to modify the translated output of said system.*" However, **Vos** discloses:

- "*...whereby said linker uses said selected data to modify the translated output of said system.*" (E.g., see Fig. 1 & Page 2, lines 21-27), wherein the linker uses the linker command file, configuration object file, and support library to modify the object code in accordance with the prototype processor system's memory.

Hadjiyiannis and **Vos** are analogous art because they are both concerned with the same field of endeavor, namely, an architecture that is modifiable by input and adapts source code to such input and correspondingly outputs machine language. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to utilize **Vos'** linker in **Hadjiyiannis's** system of claim **1** as an alternate method to implement architectural specifications. The motivation for doing so would have been to have a design that may be more efficient for a particular objective. See claim **3** for the remaining limitation.

In regard to claim **5**, **Hadjiyiannis** discloses:

- - "...providing *a descriptor file containing information descriptive of the instruction set of said target microprocessor...*" (E.g., see Fig. 1 & Section I, Paragraph C, lines 1-9), wherein the Architecture Synthesis

System comprises the machine description including an instruction set specification and some architectural information.

- "...*translating assembly language instructions into machine language wherein the translation step comprises... acquiring data from said descriptor file...*" (E.g., see Fig. 1 & Section III, Paragraph 1, lines 2-11), wherein the Architecture Synthesis System transmits the ISDL description to the compiler. The compiler outputs machine specific assembly code, which is translated to machine language via the automatic assembler generated by the ISDL description. Thus, the translation device comprises the Architecture Synthesis System, the compiler, and the ISDL description. Furthermore, the compiler fetches the ISDL description from the Architecture Synthesis System.
- "...*constraining the ...machine language to conform to the architecture of said instruction set...*" (E.g., see Fig. 1 & Section III, Paragraph 2), wherein **Hadjiyiannis** teaches that "The compiler can therefore avoid generating invalid instructions by ensuring that each instruction meets these constraints".
- "...*thereby assembling the machine language program for the target microprocessor...*" (E.g., see Fig. 1 & Section III, Paragraph 1, lines 2-11), wherein a binary file (machine language program) is assembled for the target microprocessor.

But **Hadjiyiannis** does not expressly disclose "...*directly transliterating the assembly language instructions to machine language...*" Instead **Hadjiyiannis** teaches receiving specific details of the instruction set and constraining the assembly language via translation, thereby constraining the machine language as well, to the architecture of said instruction set. Thus, it would have been obvious to one of ordinary skill in the art, to constrain the machine language to conform to the architecture of said instruction set, instead of restraining the assembly language to conform to the architecture, and thereby restraining the machine language, as the two methods produce the same result. See claim **1** for the remaining limitations.

In regard to claim **6**, **Hadjiyiannis** discloses a method as described in claim **5** above, and furthermore discloses:

- "...*wherein the descriptor file contains syntax information for each instruction of the instruction set...*" (E.g., see Section III, Paragraph 4), wherein the six sections of the ISDL are listed and later further described along with their syntax definitions.
- "...*and constraining step comprises constraining each assembly language instruction using said syntax information.*" (E.g., see Section I, Paragraph C, lines 1-6), wherein the machine description contains the syntax information and the code generator produces code based on that information.

In regard to claim **7**, claim **7** is a method version of the previously disclosed claims **1, 2 and 3**. **Hadjiyiannis** discloses the system of claims **1 and 2** as described

above, correspondingly meeting the limitations as applied to claim **7**. But **Hadjiyiannis** does not disclose expressly the limitations of claim **3**. However, **Vos** discloses the limitations of claim **3** as described above. Thus, the limitations in claim **7** are met as disclosed in the respective above claims.

In regard to claim **8**, claim **8** is a method version of claim **1** with further limitations. **Hadjiyiannis** discloses the system of claim **1** as described above. Furthermore, **Hadjiyiannis** discloses:

- "*...thereby preparing the program executable on the microprocessor.*" (E.g., see Fig. 1 & Section III, Paragraph 1, lines 2-11), wherein a binary file (machine language program) is assembled for the target microprocessor.

But **Hadjiyiannis** does not disclose expressly "*...providing plural program modules, at least one of said modules having one or more instructions including external symbols, wherein external symbols have values which cannot be determined without reference to another program module...*". Furthermore, **Hadjiyiannis** does not disclose expressly "...and further comprising binding external symbols to addresses using data selected from said descriptor file." However, **Vos** discloses:

- "*...providing plural program modules, at least one of said modules having one or more instructions including external symbols, wherein external symbols have values which cannot be determined without reference to another program module...*" (E.g., see Fig. 1, blocks 10, 12, 14, 16, 18 and 20 & Page 2, lines 15-19), wherein the Pascal

Object File has one or more instructions including external symbols, which are determined by reference to the support library.

- "...*and further comprising binding external symbols to addresses using data selected from said descriptor file*." (E.g., see Fig. 1 & Page 2, lines 2-15), wherein the generated Pascal code comprising symbols are binded to addresses of the processor system.

Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to utilize **Vos'** program modules and binding in **Hadjiyiannis and Edmond's** system of claim **1** as an alternate method to implement architectural specifications. The motivation for doing so would have been to have a simpler design for a particular system, where **Vos'** method may be more efficient than **Hadjiyiannis's** for a particular objective. See claim **5** for the remaining limitations.

In regard to claim **12**, **Edmonds** further discloses:

- "...wherein a decoding function is provided, the decoding function is used to check for at least one error in the assembly language instructions." (E.g., see Fig. 1 & "Assembler Facilities"), wherein the system decoding hardware determines that two functions have contiguous addresses, yet they are both longer than one cycle, then subsequent code must be put elsewhere and linked by a jump.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John J. Romano whose telephone number is (571) 272-3872. The examiner can normally be reached on 8-5:30, M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JJR

TUAN DAM
SUPERVISORY PATENT EXAMINER